Playing Stochastically in Weighted (Timed) Games to Emulate Memory

Julie Parreaux
Benjamin Monmege   Pierre-Alain Reynier

Aix–Marseille Université

GT Vérif meeting
November 18, 2021

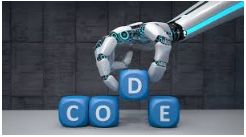
Motivation : game theory for synthesis

Classic approach

Check the correctness of a system

Game theory

Interaction between two antagonistic agents : environment and controller

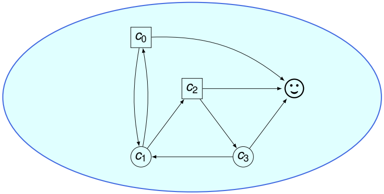


Code synthesis

Correct by construction : synthesis of controller

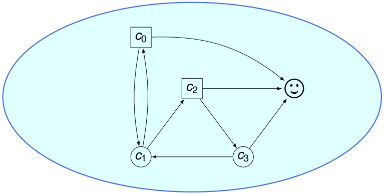
Different classes of games

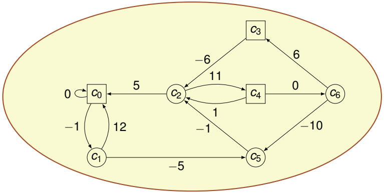
Qualitative games



Different classes of games

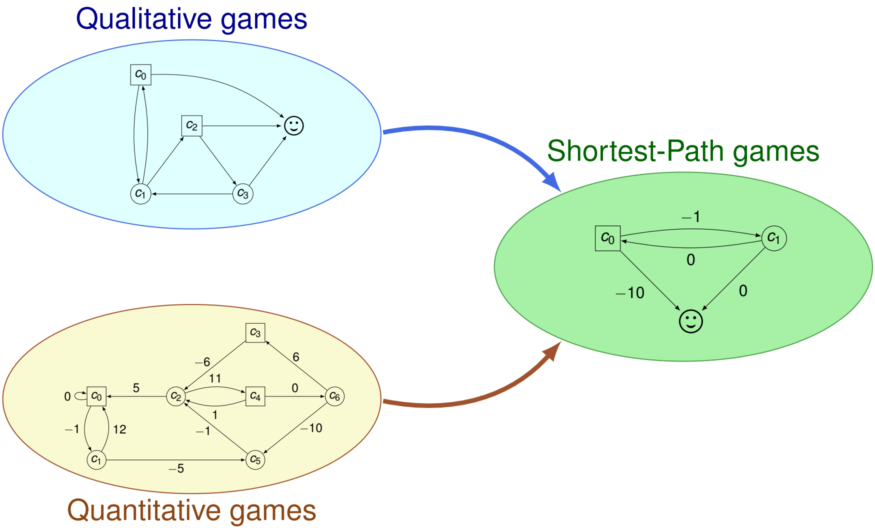
Qualitative games



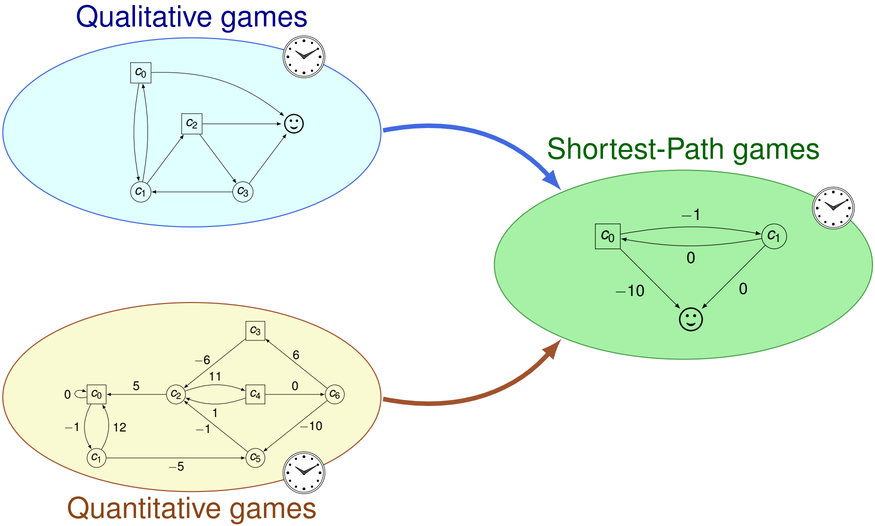


Quantitative games

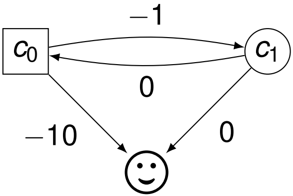
Different classes of games



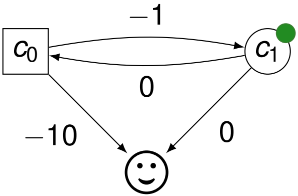
Different classes of games



Shortest Path Game



Shortest Path Game

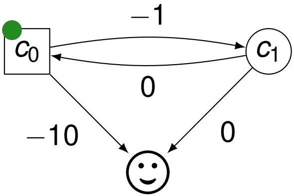


How to play?

Move a token along an edge

$$\rho = c_1$$

Shortest Path Game

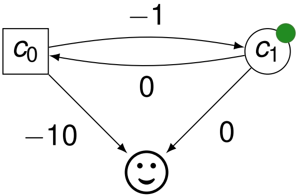


How to play?

Move a token along an edge

$$\rho = c_1 c_0$$

Shortest Path Game



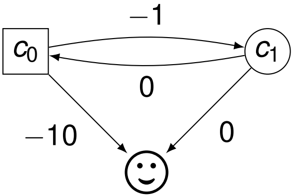
How to play?

Move a token along an edge

$$\rho = c_1 c_0 c_1$$

Shortest Path Game

◯ Min ☐ Max ☺ target



How to play?

Move a token along an edge

$$\rho = c_1 c_0 c_1 c_0 c_1 ☺$$

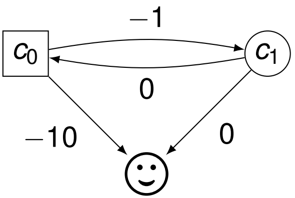
Play

Infinite path or reach the target

$\rho = (c_i)_i \in C^\omega$ $\rho = (c_i)_i ☺$

Shortest Path Game

○ Min □ Max ☺ target



How to play?

Move a token along an edge

$\rho = c_1 c_0 c_1 c_0 c_1$ ☺

Play

Infinite path or reach the target

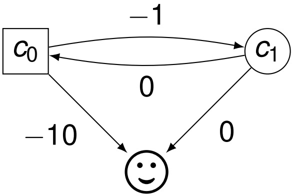
$\rho = (c_i)_i \in C^\omega$ $\rho = (c_i)_i$ ☺

Shortest Path payoff of a play ρ

$$\mathbf{SP}(\rho) = \begin{cases} \text{wt}(\rho) & \text{if } \rho \text{ reaches } ☺ \\ +\infty & \text{if } \rho \text{ does not reach } ☺ \end{cases}$$

Shortest Path Game

○ Min □ Max ☺ target



How to play?

Move a token along an edge

$$\rho = c_1 c_0 c_1 c_0 c_1 ☺$$

$$\mathbf{SP}(\rho) = 0 + (-1) + 0 + (-1) + 0 = -2$$

Play

Infinite path or reach the target

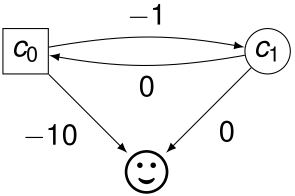
$\rho = (c_i)_i \in C^\omega$ $\qquad$ $\rho = (c_i)_i$ ☺

Shortest Path payoff of a play ρ

$$\mathbf{SP}(\rho) = \begin{cases} \mathrm{wt}(\rho) & \text{if } \rho \text{ reaches ☺} \\ +\infty & \text{if } \rho \text{ does not reach ☺} \end{cases}$$

Shortest Path Game

○ Min □ Max ☺ target



How to play?

Move a token along an edge

$$\rho = c_1 c_0 c_1 c_0 c_1 ☺$$
$$\mathbf{SP}(\rho) = 0 + (-1) + 0 + (-1) + 0 = -2$$

Play

Infinite path or reach the target

$\rho = (c_i)_i \in C^\omega$ $\rho = (c_i)_i ☺$

Objectives

Max maximise the payoff

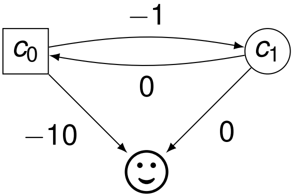
Min minimise the payoff

Shortest Path payoff of a play ρ

$$\mathbf{SP}(\rho) = \begin{cases} \mathrm{wt}(\rho) & \text{if } \rho \text{ reaches ☺} \\ +\infty & \text{if } \rho \text{ does not reach ☺} \end{cases}$$

Shortest Path Game

○ Min □ Max ☺ target



How to play?

Move a token along an edge

$$\rho = c_1 c_0 c_1 c_0 c_1 ☺$$
$$\mathbf{SP}(\rho) = 0 + (-1) + 0 + (-1) + 0 = -2$$

Play

Infinite path or reach the target

$\rho = (c_i)_i \in C^\omega$ $\qquad$ $\rho = (c_i)_i ☺$

Objectives

Max maximise the payoff

Min minimise the payoff

Shortest Path payoff of a play ρ

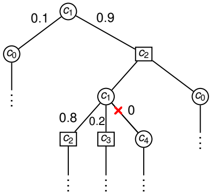
$$\mathbf{SP}(\rho) = \begin{cases} \mathrm{wt}(\rho) & \text{if } \rho \text{ reaches ☺} \\ +\infty & \text{if } \rho \text{ does not reach ☺} \end{cases}$$

Hypothesis

Divergent games: no zero cycle

Zoology of strategies

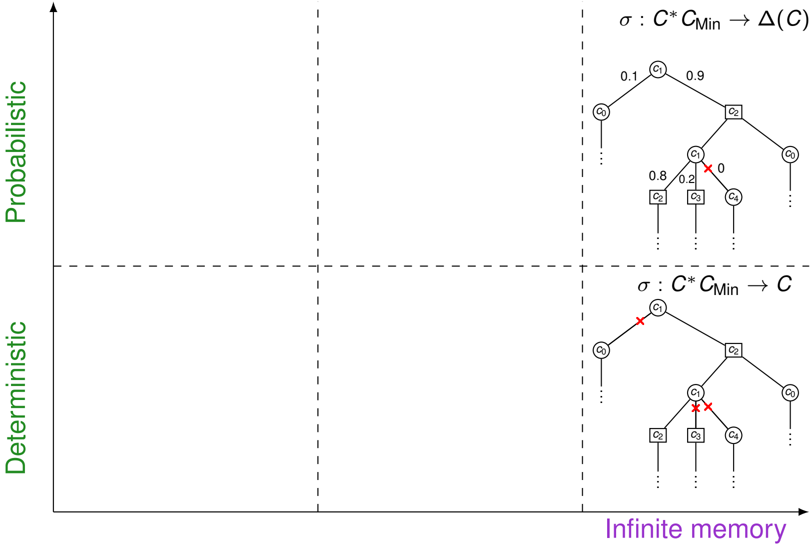
$\sigma : C^* C_{\mathrm{Min}} \to \Delta(C)$



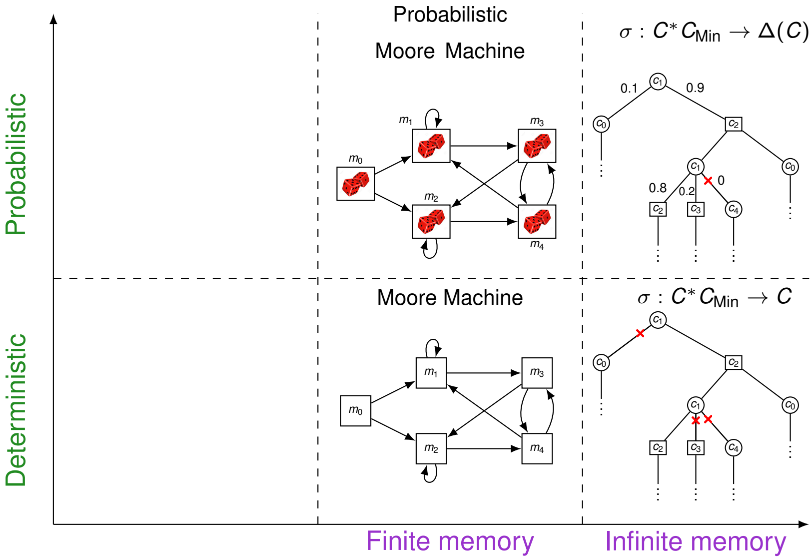
Probabilistic

Infinite memory

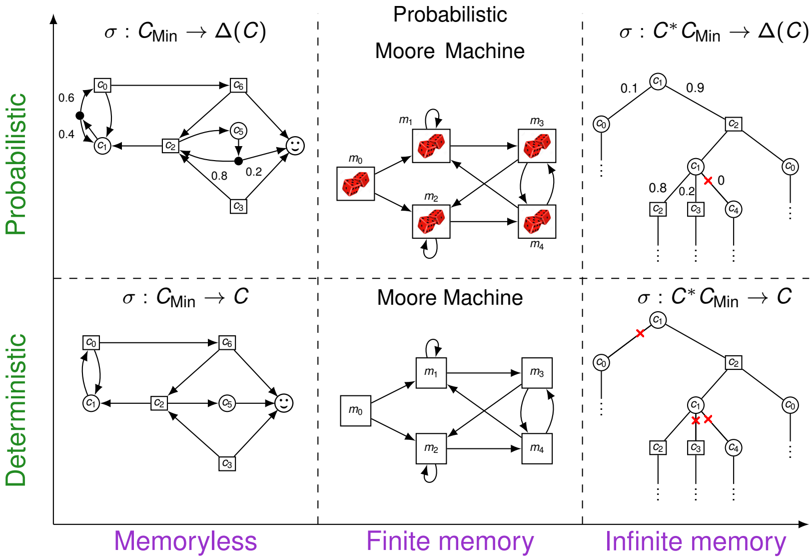
Zoology of strategies



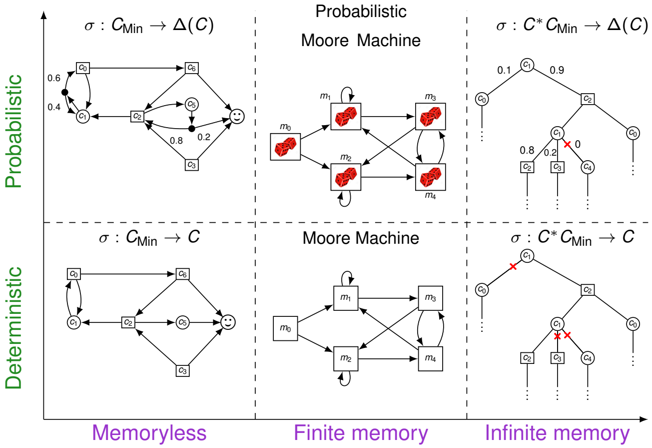
Zoology of strategies



Zoology of strategies

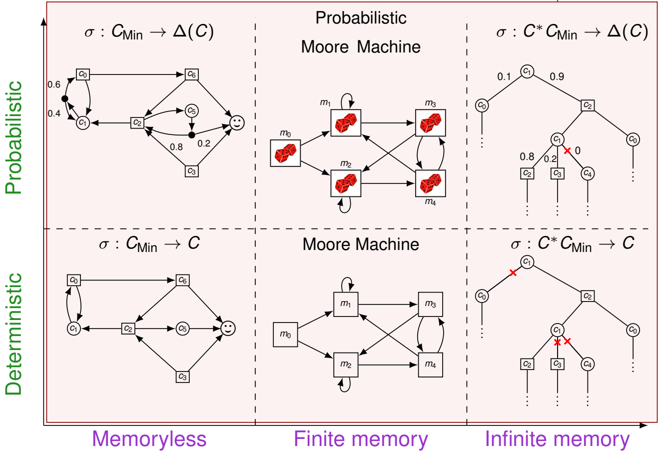


Stochastic values

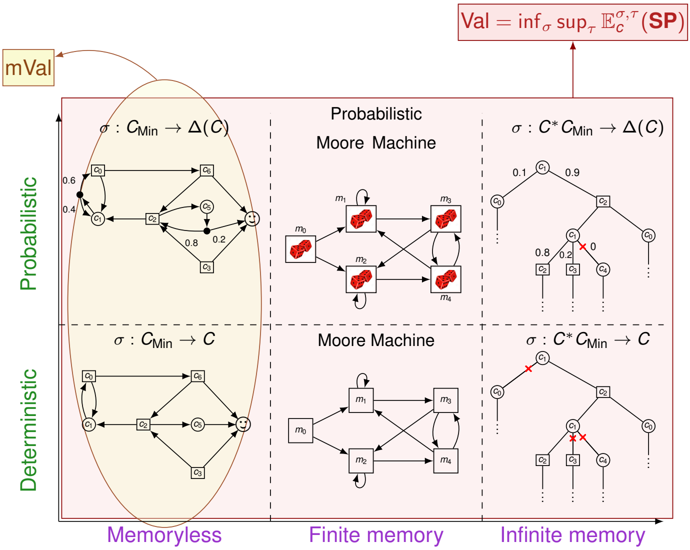


Stochastic values

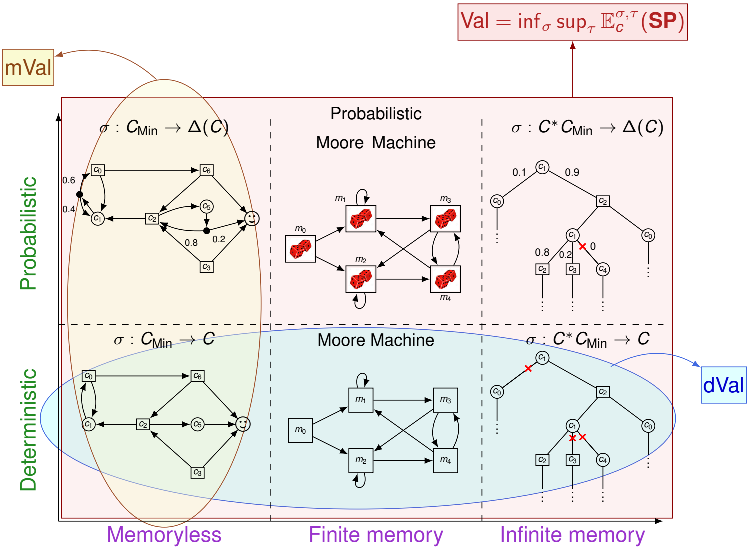
$\mathrm{Val} = \inf_\sigma \sup_\tau \mathbb{E}_c^{\sigma,\tau}(\mathbf{SP})$



Stochastic values



Stochastic values



Contribution

$$\text{dVal} \quad = \quad \text{Val} \quad = \quad \text{mVal}$$

Contribution

Trade-off between memory and randomness

$$\mathsf{dVal} \quad = \quad \mathsf{Val} \quad = \quad \mathsf{mVal}$$

Contribution

Trade-off between memory and randomness

- Stochastic games with qualitative objectives

$$\mathrm{dVal} \quad = \quad \mathrm{Val} \quad = \quad \mathrm{mVal}$$

Trading Memory for Randomness, K. Chatterjee, L. Alfaro and T. Henzinger, 2004, QEST

Contribution

Trade-off between memory and randomness

- Stochastic games with qualitative objectives
- Reachability Timed Games

$$\text{dVal} = \text{Val} = \text{mVal}$$

Trading Memory for Randomness, K. Chatterjee, L. Alfaro and T. Henzinger, 2004, QEST

Trading Infinite Memory for Uniform Randomness in Timed Games, K. Chatterjee, T. Henzinger and S. Vinayak, 2008, HSCC

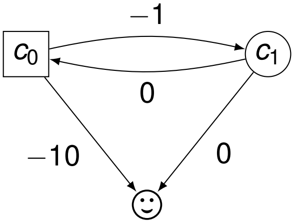
Deterministic strategies: Min needs memory

σ Min

τ Max

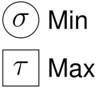
Value

$$\mathrm{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{SP}(\mathrm{Play}(c, \sigma, \tau))$$



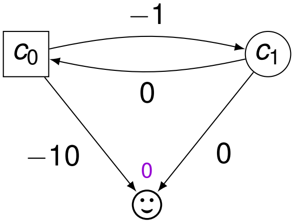
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory



Value

$$\mathsf{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{SP}(\mathsf{Play}(c, \sigma, \tau))$$



Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

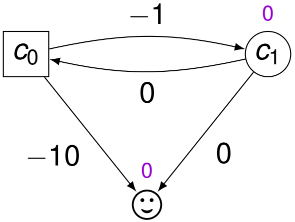
Deterministic strategies: Min needs memory

σ Min

τ Max

Value

$$\mathrm{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{SP}(\mathrm{Play}(c, \sigma, \tau))$$



Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

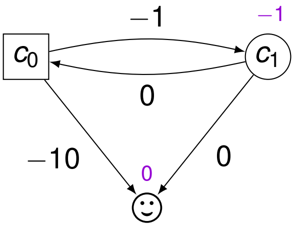
Deterministic strategies: Min needs memory

σ Min

τ Max

Value

$$\mathrm{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{SP}(\mathrm{Play}(c, \sigma, \tau))$$



Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

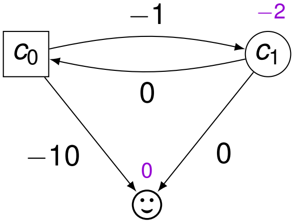
Deterministic strategies: Min needs memory

σ Min

τ Max

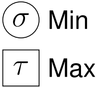
Value

$$\mathrm{dVal}(c) = \inf_\sigma \sup_\tau \mathbf{SP}(\mathrm{Play}(c, \sigma, \tau))$$



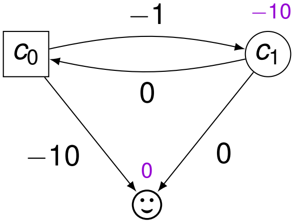
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory



Value

$$\mathsf{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{SP}(\mathsf{Play}(c, \sigma, \tau))$$



Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

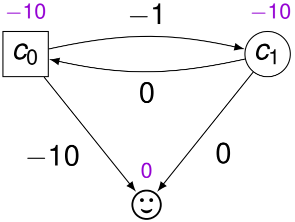
Deterministic strategies: Min needs memory

σ Min

τ Max

Value

$$\mathrm{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{SP}(\mathrm{Play}(c, \sigma, \tau))$$



Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory

σ Min

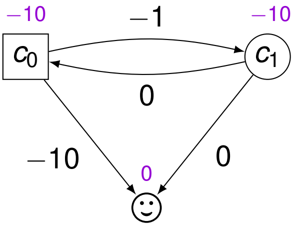
τ Max

Value

$$\mathsf{dVal}(c) = \inf_\sigma \underbrace{\sup_\tau \mathbf{SP}(\mathsf{Play}(c,\sigma,\tau))}_{\mathsf{dVal}^\sigma(c)}$$

Optimal strategy

$$\mathsf{dVal}^{\sigma^*}(c) \leqslant \mathsf{dVal}(c)$$



Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory

σ Min

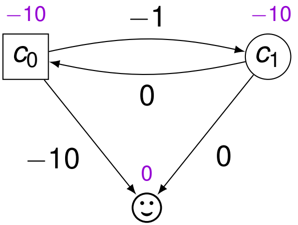
τ Max

Value

$$\mathsf{dVal}(c) = \inf_\sigma \underbrace{\sup_\tau \mathbf{SP}(\mathsf{Play}(c,\sigma,\tau))}_{\mathsf{dVal}^\sigma(c)}$$

Optimal strategy

$$\mathsf{dVal}^{\sigma^*}(c) \leqslant \mathsf{dVal}(c)$$

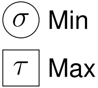


Optimal strategy for Min

An optimal strategy for Min may require finite memory.

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory

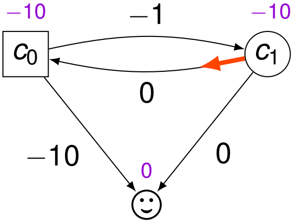


Value

$$\text{dVal}(c) = \inf_\sigma \underbrace{\sup_\tau \mathbf{SP}(\text{Play}(c, \sigma, \tau))}_{\text{dVal}^\sigma(c)}$$

Optimal strategy

$$\text{dVal}^{\sigma^*}(c) \leqslant \text{dVal}(c)$$

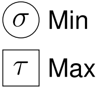


Optimal strategy for Min

An optimal strategy for Min may require finite memory.

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory

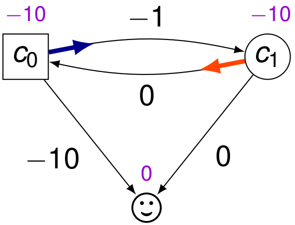


Value

$$\mathsf{dVal}(c) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathsf{Play}(c,\sigma,\tau))}_{\mathsf{dVal}^{\sigma}(c)}$$

Optimal strategy

$$\mathsf{dVal}^{\sigma^*}(c) \leqslant \mathsf{dVal}(c)$$



Optimal strategy for Min

An optimal strategy for Min may require finite memory.

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory

σ Min

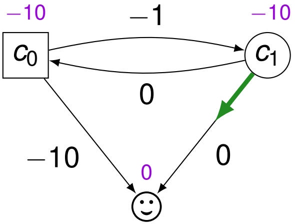
τ Max

Value

$$\mathsf{dVal}(c) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathsf{Play}(c, \sigma, \tau))}_{\mathsf{dVal}^{\sigma}(c)}$$

Optimal strategy

$$\mathsf{dVal}^{\sigma^*}(c) \leqslant \mathsf{dVal}(c)$$



Optimal strategy for Min

An optimal strategy for Min may require finite memory.

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory

σ Min

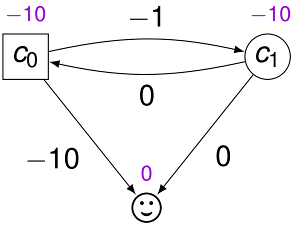
τ Max

Value

$$\mathsf{dVal}(c) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathsf{Play}(c, \sigma, \tau))}_{\mathsf{dVal}^{\sigma}(c)}$$

Optimal strategy

$$\mathsf{dVal}^{\sigma^*}(c) \leqslant \mathsf{dVal}(c)$$



Optimal strategy for Min

Switching strategy:

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory

σ Min

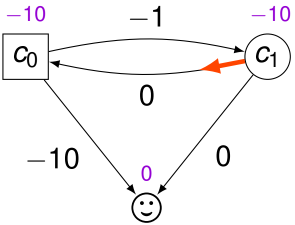
τ Max

Value

$$\mathsf{dVal}(c) = \inf_\sigma \underbrace{\sup_\tau \mathbf{SP}(\mathsf{Play}(c,\sigma,\tau))}_{\mathsf{dVal}^\sigma(c)}$$

Optimal strategy

$$\mathsf{dVal}^{\sigma^*}(c) \leqslant \mathsf{dVal}(c)$$



Optimal strategy for Min

Switching strategy:

- σ_1: reach negative cycle

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory

σ Min

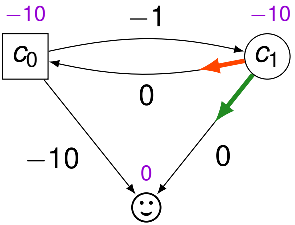
τ Max

Value

$$\mathsf{dVal}(c) = \inf_\sigma \underbrace{\sup_\tau \mathbf{SP}(\mathsf{Play}(c,\sigma,\tau))}_{\mathsf{dVal}^\sigma(c)}$$

Optimal strategy

$\mathsf{dVal}^{\sigma^*}(c) \leqslant \mathsf{dVal}(c)$



Optimal strategy for Min

Switching strategy:

- σ_1: reach negative cycle
- σ_2: reach ☺

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic strategies: Min needs memory

σ Min

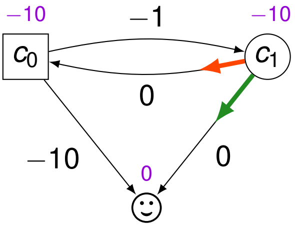
τ Max

Value

$$\mathsf{dVal}(c) = \inf_\sigma \underbrace{\sup_\tau \mathbf{SP}(\mathsf{Play}(c,\sigma,\tau))}_{\mathsf{dVal}^\sigma(c)}$$

Optimal strategy

$\mathsf{dVal}^{\sigma^*}(c) \leqslant \mathsf{dVal}(c)$



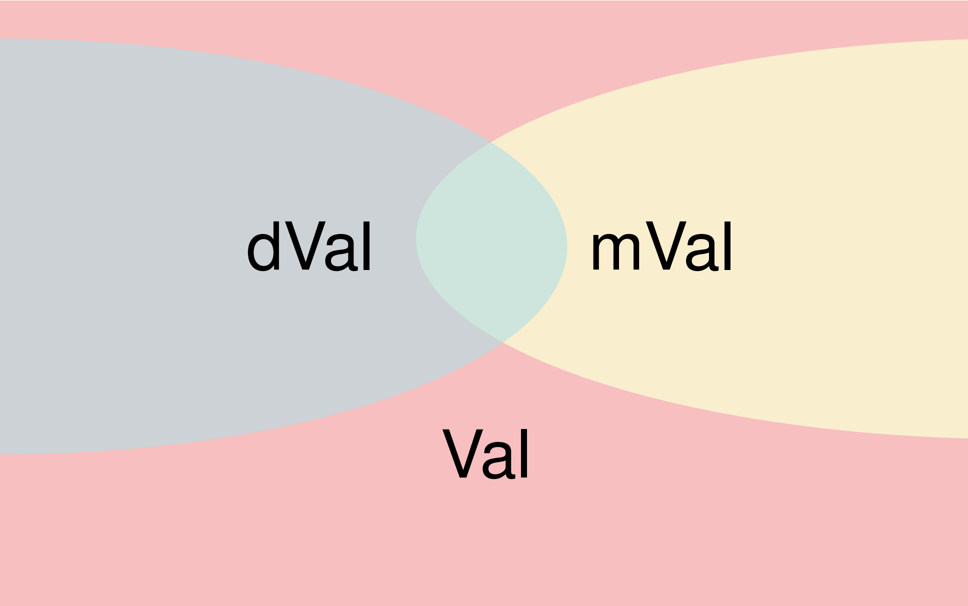
Optimal strategy for Min

Switching strategy:

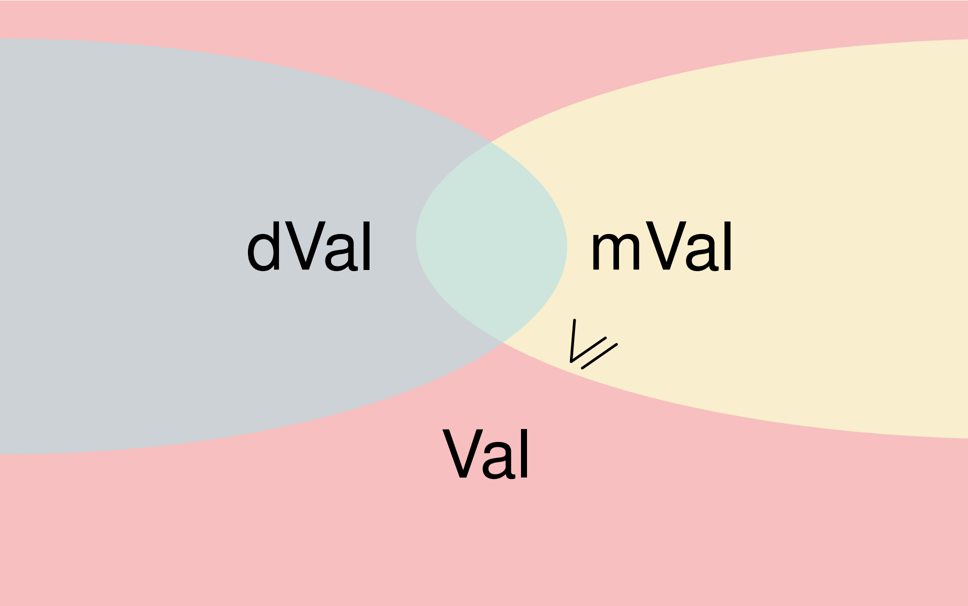
- σ_1: reach negative cycle
- σ_2: reach ☺
- K: number of turns before switch

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

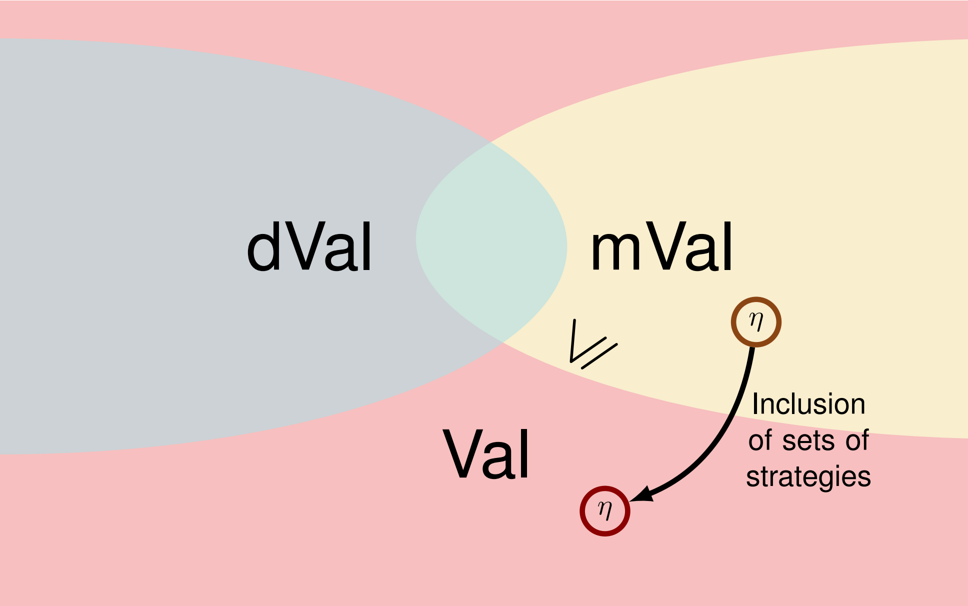
Contribution



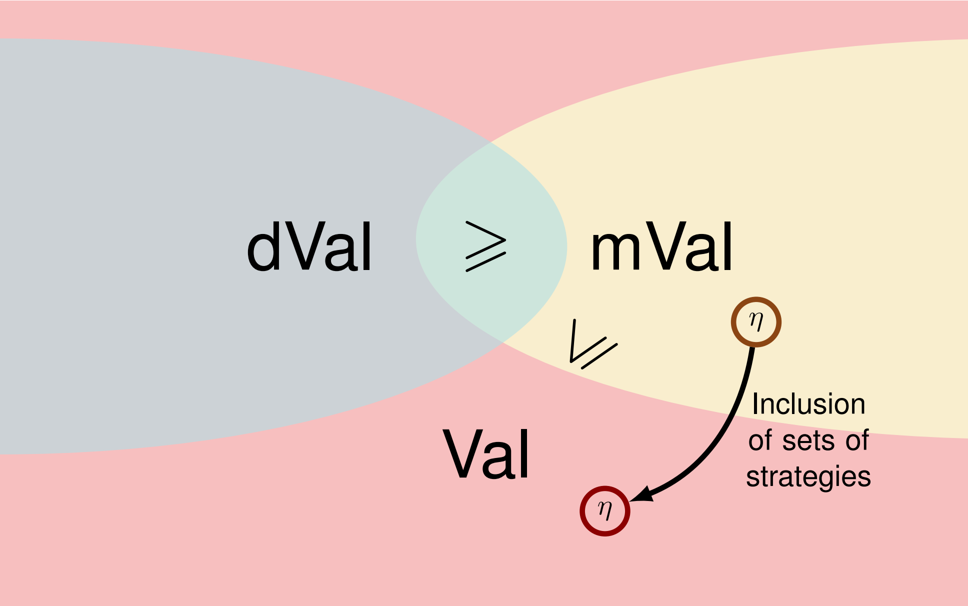
Contribution



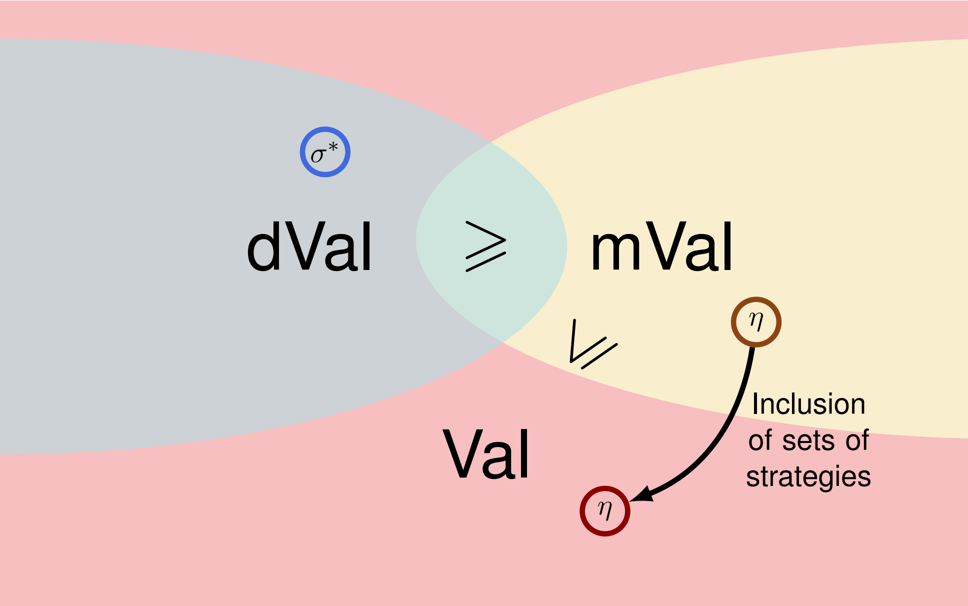
Contribution



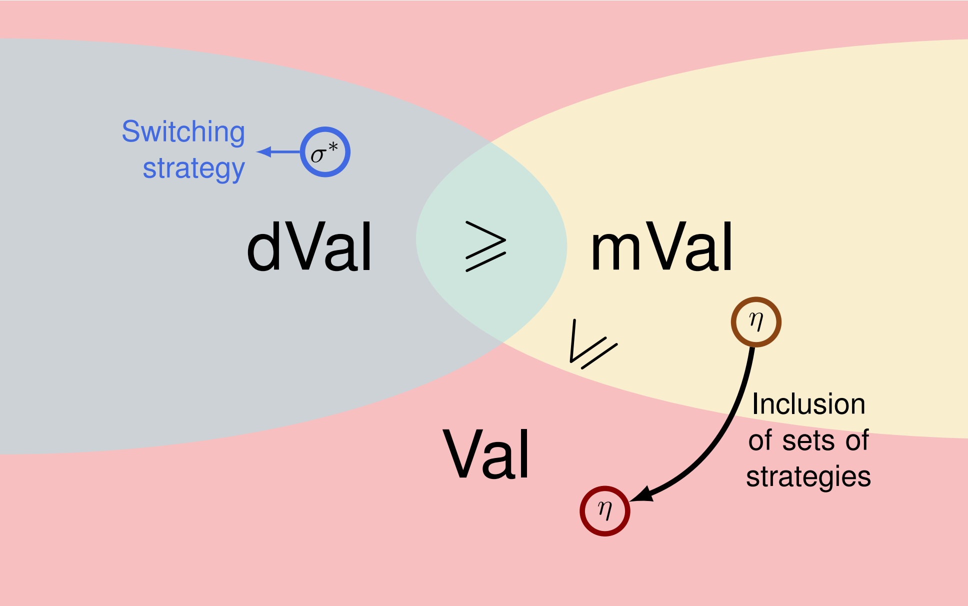
Contribution



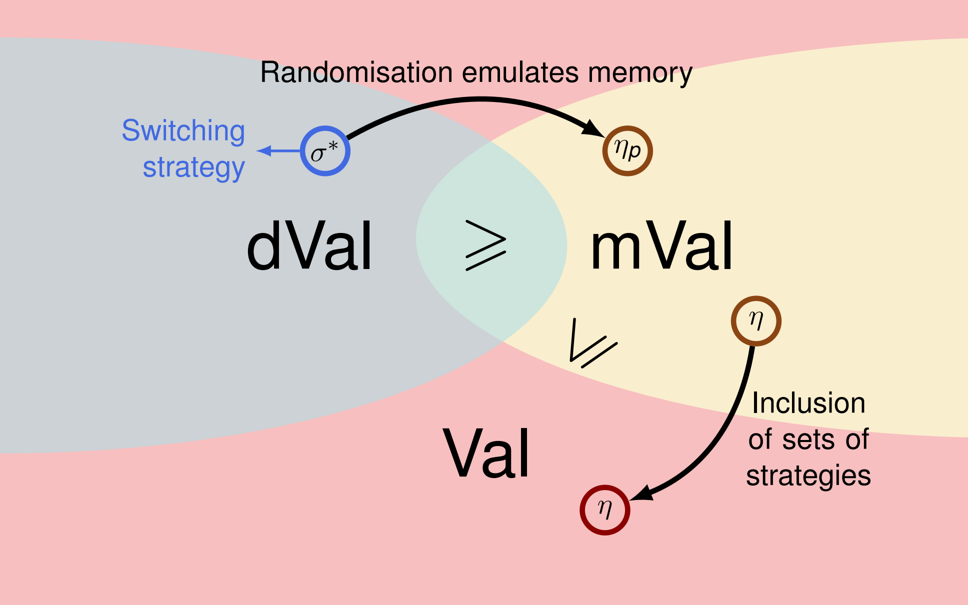
Contribution



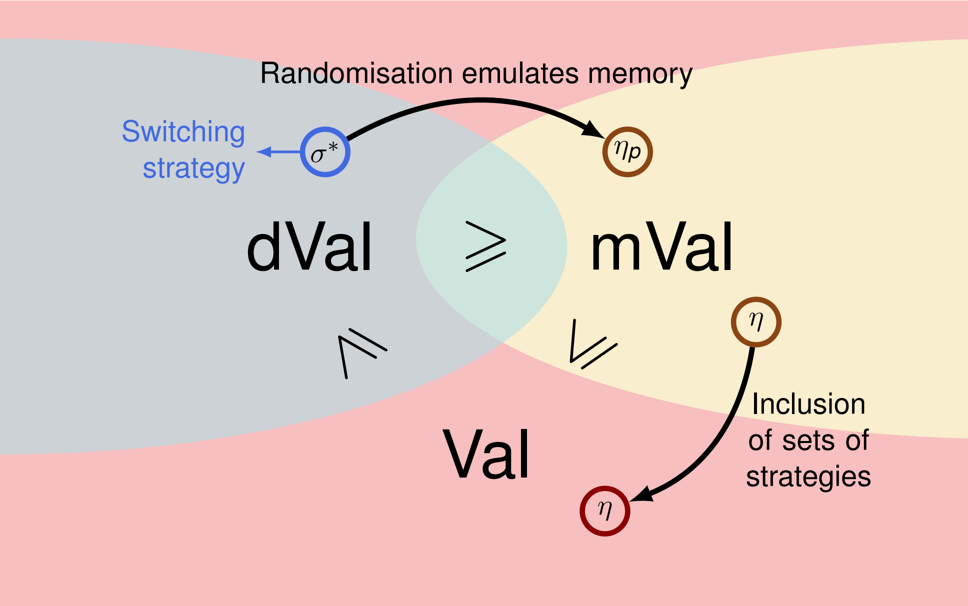
Contribution



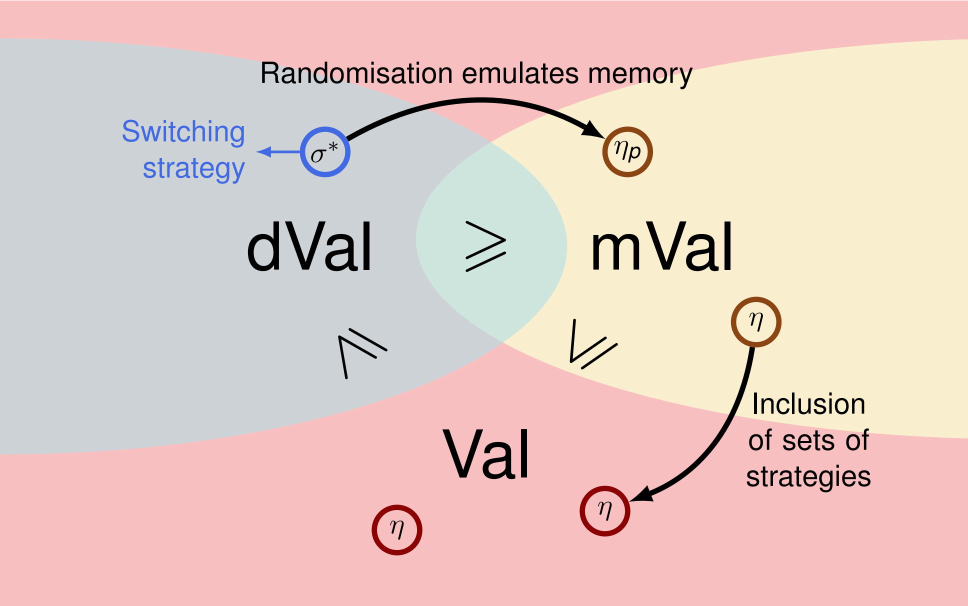
Contribution



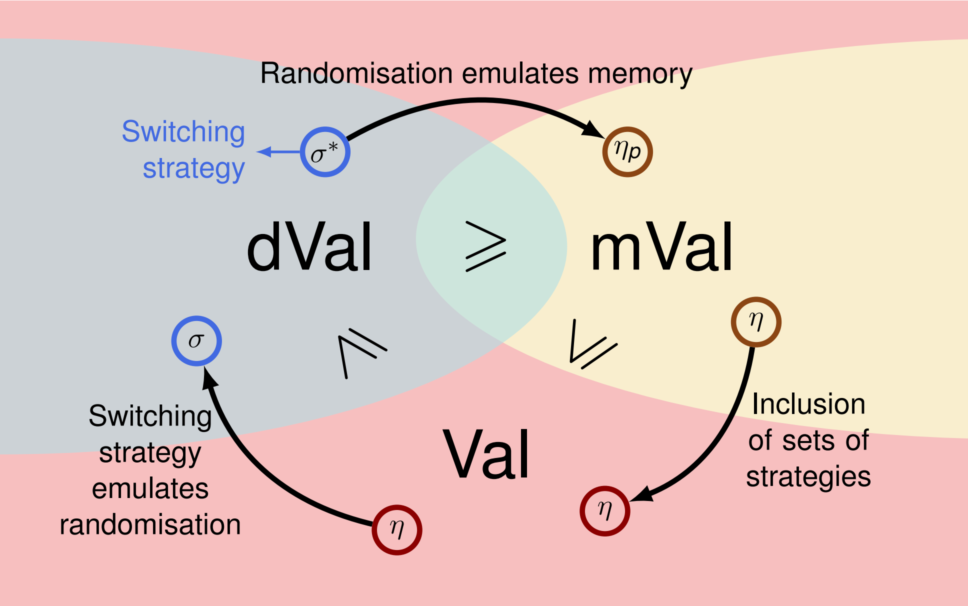
Contribution



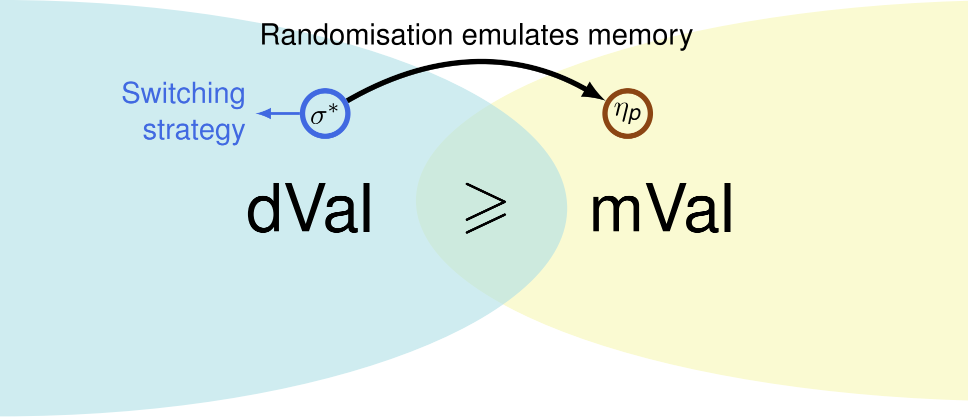
Contribution



Contribution



Contribution



Randomisation emulates memory

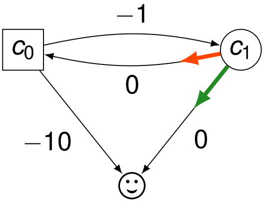
Claim

$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$

Randomisation emulates memory

Claim

$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$



Strategy η_p

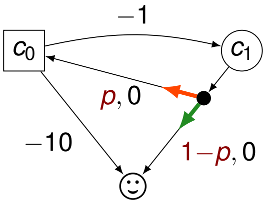
Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy,

Randomisation emulates memory

○ Min □ Max

Claim

$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$



Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

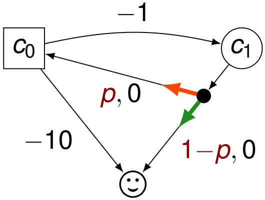
$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Randomisation emulates memory

◯ Min ☐ Max

Claim

$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$



Properties of η_p

- For all τ, $\mathbb{P}_c^{\eta_p,\tau}(\diamond☺) = 1$

Strategy η_p

Let $\langle \sigma_1, \sigma_2, K\rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

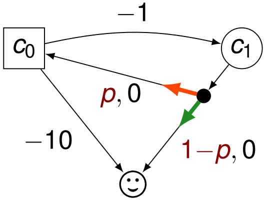
$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Randomisation emulates memory

◯ Min ☐ Max

Claim

$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathsf{mVal}^{\eta_p}(c) \leqslant \mathsf{dVal}(c)$$



Properties of η_p

- For all τ, $\mathbb{P}_c^{\eta_p,\tau}(\diamond ☺) = 1$
- For all τ, $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) < \infty$

Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

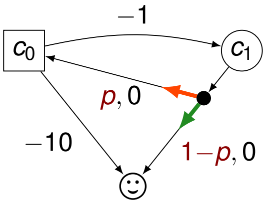
$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Randomisation emulates memory

Min Max

Claim

$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$



Properties of η_p

- For all τ, $\mathbb{P}_c^{\eta_p,\tau}(\diamond☺) = 1$
- For all τ, $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) < \infty$
- Max has a best response memoryless deterministic strategy: τ

Strategy η_p

Let $\langle \sigma_1, \sigma_2, K\rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

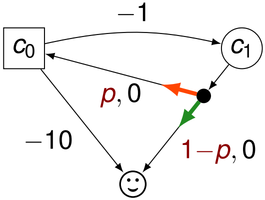
$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Randomisation emulates memory

○ Min ☐ Max

Claim

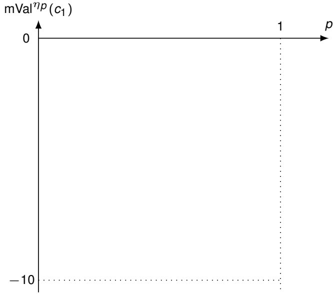
$$\forall c,\ \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$



Properties of η_p

- For all τ, $\mathbb{P}_c^{\eta_p,\tau}(\diamond ☺) = 1$
- For all τ, $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) < \infty$
- Max has a best response memoryless deterministic strategy: τ

Computation of $\mathrm{mVal}^{\eta_p}(c_1)$



Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

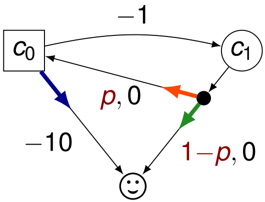
$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Randomisation emulates memory

○ Min □ Max

Claim

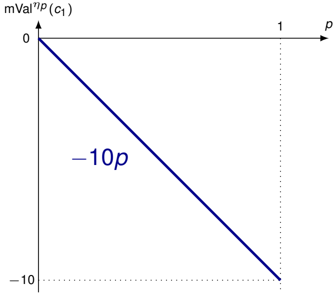
$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$



Properties of η_p

- For all τ, $\mathbb{P}_c^{\eta_p,\tau}(\diamond ☺) = 1$
- For all τ, $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) < \infty$
- Max has a best response memoryless deterministic strategy: τ

Computation of $\mathrm{mVal}^{\eta_p}(c_1)$



Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

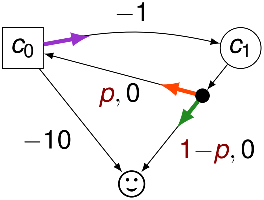
$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Randomisation emulates memory

○ Min □ Max

Claim

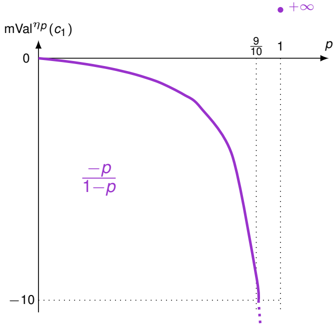
$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$



Properties of η_p

- For all τ, $\mathbb{P}_c^{\eta_p,\tau}(\diamond☺) = 1$
- For all τ, $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) < \infty$
- Max has a best response memoryless deterministic strategy: τ

Computation of $\mathrm{mVal}^{\eta_p}(c_1)$



Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

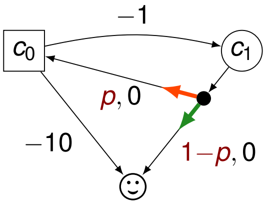
$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Randomisation emulates memory

Min Max

Claim

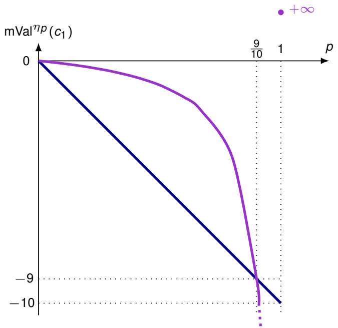
$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$



Properties of η_p

- For all τ, $\mathbb{P}_c^{\eta_p,\tau}(\diamond ☺) = 1$
- For all τ, $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) < \infty$
- Max has a best response memoryless deterministic strategy: τ

Computation of $\mathrm{mVal}^{\eta_p}(c_1)$



Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

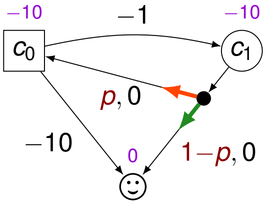
$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Randomisation emulates memory

○ Min □ Max

Claim

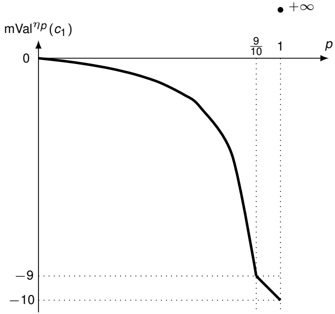
$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$



Properties of η_p

- For all τ, $\mathbb{P}_c^{\eta_p,\tau}(\diamond☺) = 1$
- For all τ, $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) < \infty$
- Max has a best response memoryless deterministic strategy: τ

Computation of $\mathrm{mVal}^{\eta_p}(c_1)$



Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0,1)$,

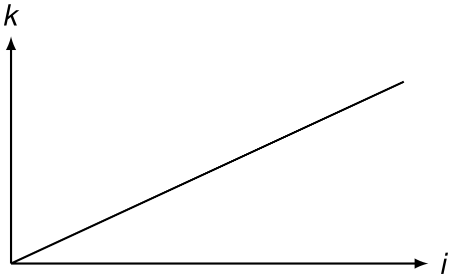
$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Computation of the expectation $\mathbb{E}_{c}^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_{c}^{\eta_p,\tau}(\mathbf{SP}) = \sum_{\rho} \mathbf{SP}(\rho)\,\mathbb{P}(\rho)$$

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) = \sum_{\rho} \mathbf{SP}(\rho)\,\mathbb{P}(\rho) = \quad + \quad +$$

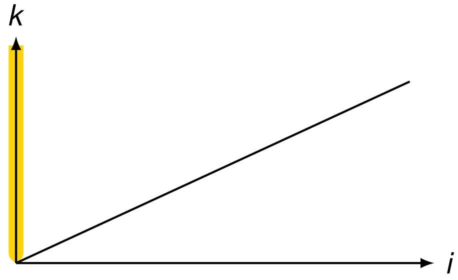


k size of play reaching the target

i number of choices given by σ_2

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) = \sum_{\rho} \mathbf{SP}(\rho)\,\mathbb{P}(\rho) = \quad + \quad +$$



Yellow zone

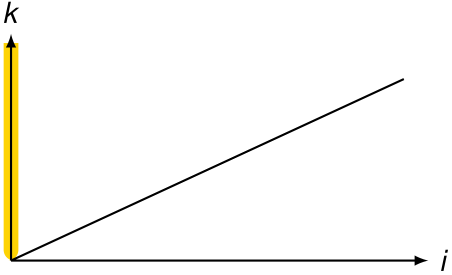
All plays conforming to σ_1

k size of play reaching the target

i number of choices given by σ_2

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) = \sum_\rho \mathbf{SP}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \quad +$$



k size of play reaching the target

i number of choices given by σ_2

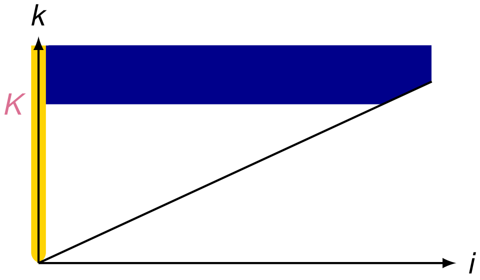
Yellow zone

All plays conforming to σ_1

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}$

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) = \sum_{\rho} \mathbf{SP}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} +$$



k size of play reaching the target

i number of choices given by σ_2

Blue zone

Plays with many negative cycles

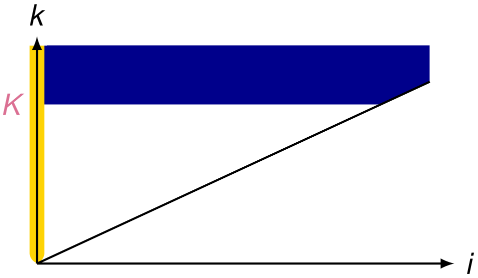
Yellow zone

All plays conforming to σ_1

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}$

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) = \sum_{\rho} \mathbf{SP}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} +$$



k size of play reaching the target

i number of choices given by σ_2

Blue zone

Plays with many negative cycles

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}$

Yellow zone

All plays conforming to σ_1

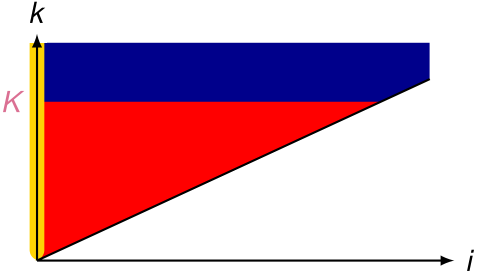
$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}$

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) = \sum_{\rho} \mathbf{SP}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} + \mathbb{E}$$

Red zone

Rest of plays



Blue zone

Plays with many negative cycles

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}$

Yellow zone

All plays conforming to σ_1

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}$

k size of play reaching the target

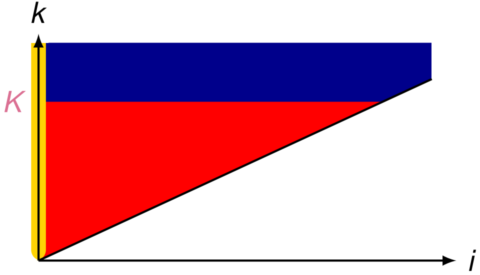
i number of choices given by σ_2

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) = \sum_{\rho} \mathbf{SP}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} + \mathbb{E}$$

Red zone

Rest of plays $\quad \mathbb{E} \xrightarrow[\substack{p\to 1\\ p<1}]{} 0$



k size of play reaching the target

i number of choices given by σ_2

Blue zone

Plays with many negative cycles

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}$

Yellow zone

All plays conforming to σ_1

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}$

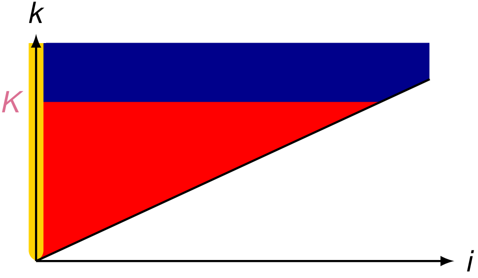
Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) = \sum_{\rho} \mathbf{SP}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} + \mathbb{E}$$

Red zone

Rest of plays

$$\mathbb{E} \underset{\substack{p \to 1 \\ p<1}}{\longrightarrow} 0$$



k size of play reaching the target

i number of choices given by σ_2

Blue zone

Plays with many negative cycles

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}$

$$\lim_{\substack{p \to 1 \\ p<1}} \mathbb{E} + \mathbb{E} \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2 K \rangle}$$

Yellow zone

All plays conforming to σ_1

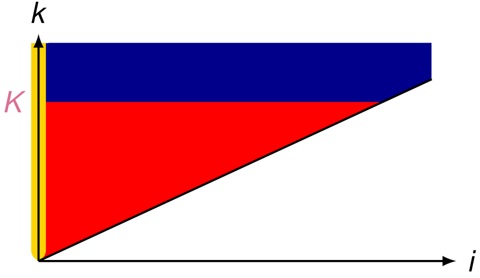
$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}$

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{SP}) = \sum_\rho \mathbf{SP}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} + \mathbb{E} \Rightarrow \lim_{\substack{p\to 1\\ p<1}} \mathbb{E}_c^{\eta_p,\tau} \leqslant \mathrm{dVal}^{\langle \sigma_1,\sigma_2,K\rangle}$$

Red zone

Rest of plays $\quad \mathbb{E} \underset{\substack{p\to 1\\ p<1}}{\longrightarrow} 0$



k size of play reaching the target

i number of choices given by σ_2

Blue zone

Plays with many negative cycles

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1,\sigma_2,K\rangle}$

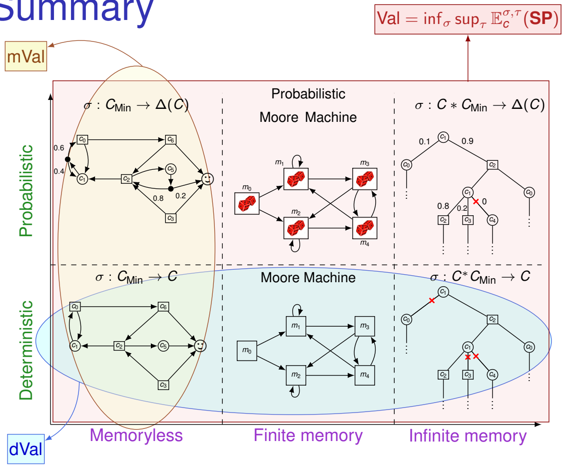
$$\lim_{\substack{p\to 1\\ p<1}} \mathbb{E} + \mathbb{E} \leqslant \mathrm{dVal}^{\langle \sigma_1,\sigma_2 K\rangle}$$

Yellow zone

All plays conforming to σ_1

$\mathbf{SP}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1,\sigma_2,K\rangle}$

Summary



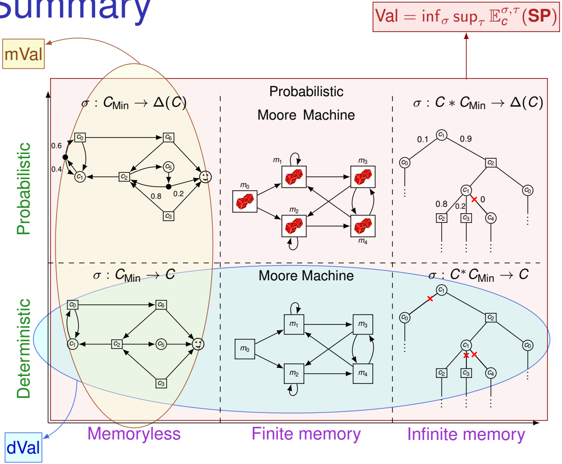
For which classes of games?

- Divergent finite shortest path games

Theorem

$$\text{Val} = \text{dVal} = \text{mVal}$$

Summary



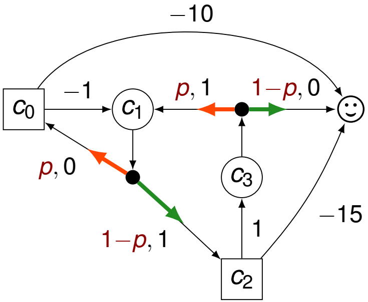
For which classes of games?

- Finite shortest path games

Theorem

Val = dVal = mVal

Extension to shortest path games



Claim

$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$

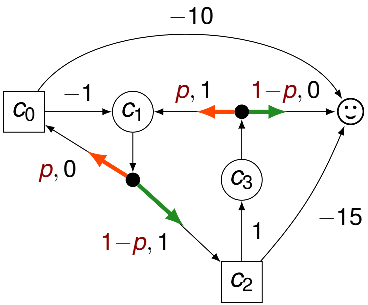
Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Extension to shortest path games

◯ Min ☐ Max



Claim

$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathsf{mVal}^{\eta_p}(c) \leqslant \mathsf{dVal}(c)$$

Problem

Presence of non-negative cycles

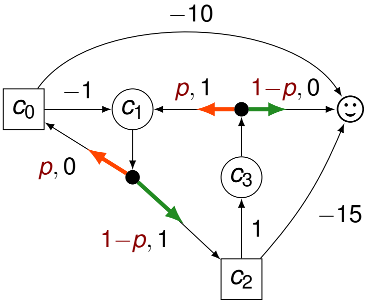
Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Extension to shortest path games

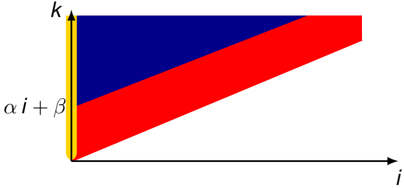
◯ Min ☐ Max



Claim

$$\forall c, \lim_{\substack{p \to 1 \\ p < 1}} \mathrm{mVal}^{\eta_p}(c) \leqslant \mathrm{dVal}(c)$$

Need a new partition



Problem

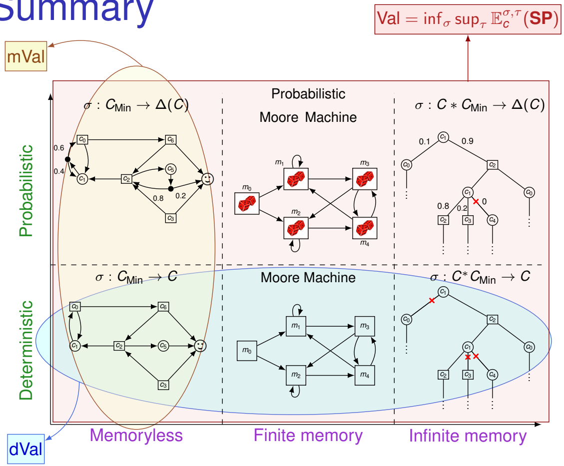
Presence of non-negative cycles

Strategy η_p

Let $\langle \sigma_1, \sigma_2, K \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\eta_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Summary



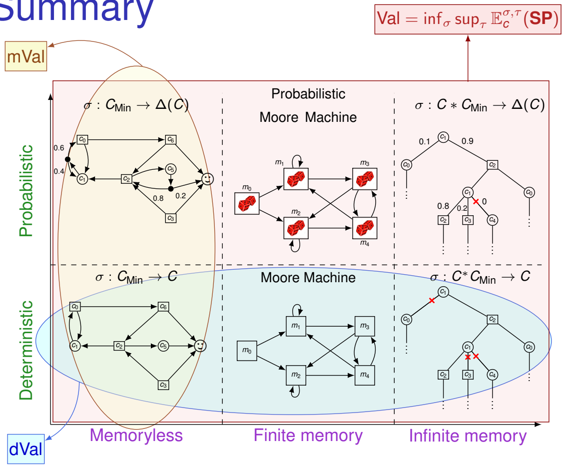
For which classes of games?

- Finite shortest path games

Theorem

$\mathsf{Val} = \mathsf{dVal} = \mathsf{mVal}$

Summary



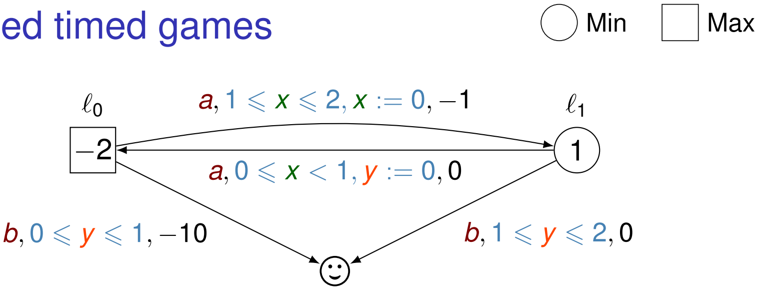
For which classes of games?

- Finite shortest path games
- Divergent weighted timed games

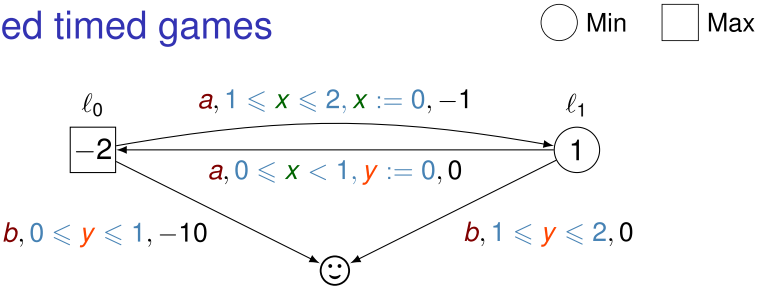
Theorem

$\mathsf{Val} = \mathsf{dVal} = \mathsf{mVal}$

Weighted timed games



Weighted timed games

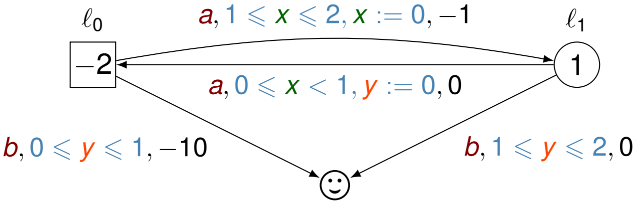


Play ρ

$$\left(\ell_1, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right) \xrightarrow{a,\ 0.5} \left(\ell_0, \begin{pmatrix} 0.5 \\ 0 \end{pmatrix}\right) \xrightarrow{a,\ 1.25} \left(\ell_1, \begin{pmatrix} 0 \\ 1.25 \end{pmatrix}\right) \xrightarrow{b,\ 1/3} \left(☺, \begin{pmatrix} 1/3 \\ 19/12 \end{pmatrix}\right)$$

Weighted timed games

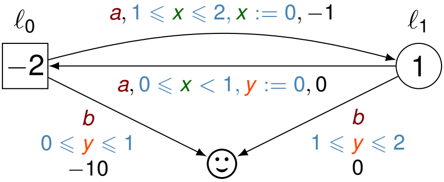
◯ Min ☐ Max



Play ρ

$$(\ell_1, \begin{pmatrix} 0 \\ 0 \end{pmatrix}) \xrightarrow[1 \times 0.5 + 0]{a,\ 0.5} (\ell_0, \begin{pmatrix} 0.5 \\ 0 \end{pmatrix}) \xrightarrow[-2 \times 1.25 - 1]{a,\ 1.25} (\ell_1, \begin{pmatrix} 0 \\ 1.25 \end{pmatrix}) \xrightarrow[1 \times \frac{1}{3} + 0]{b,\ 1/3} (☺, \begin{pmatrix} 1/3 \\ 19/12 \end{pmatrix}) \rightsquigarrow -\frac{8}{3}$$

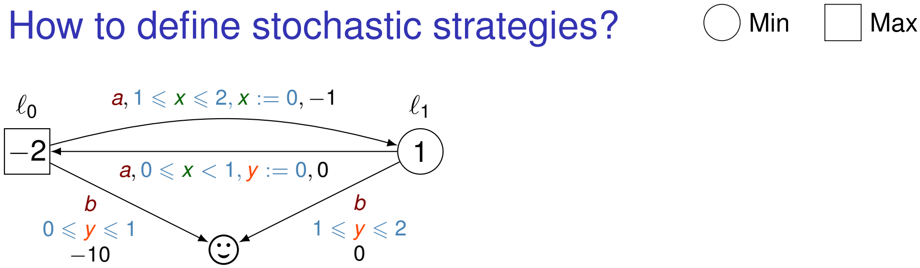
How to define stochastic strategies?



Deterministic strategy

Choose an edge and a delay

How to define stochastic strategies?



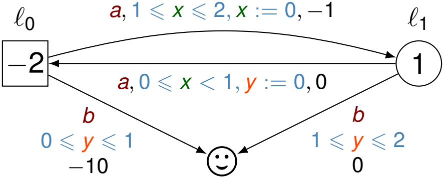
Deterministic strategy

Choose an edge and a delay

In $(\ell_1, (0, 0))$

Choose a with $t = \frac{1}{3}$

How to define stochastic strategies?



Probabilistic strategy

Distribution over possible choices

Deterministic strategy

Choose an edge and a delay

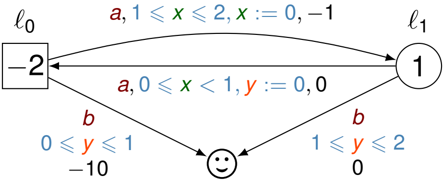
In $(\ell_1, (0, 0))$

Choose a with $t = \frac{1}{3}$

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

How to define stochastic strategies?

○ Min □ Max



Probabilistic strategy

Distribution over possible choices

1. Edge *a*: finite distribution

Deterministic strategy

Choose an edge and a delay

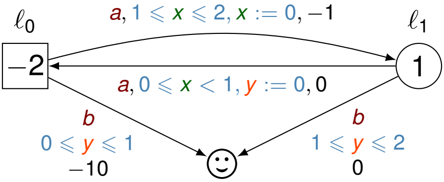
In $(\ell_1, (0, 0))$

Choose *a* with $t = \frac{1}{3}$

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

How to define stochastic strategies?

◯ Min   □ Max



Probabilistic strategy

Distribution over possible choices

1. Edge a: finite distribution
2. Delay for a: infinite distribution

Deterministic strategy

Choose an edge and a delay

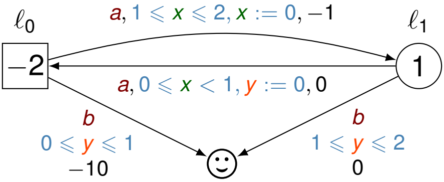
In $(\ell_1, (0, 0))$

Choose a with $t = \frac{1}{3}$

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

How to define stochastic strategies?

○ Min ☐ Max



Deterministic strategy

Choose an edge and a delay

In $(\ell_1, (0, 0))$

Choose *a* with $t = \frac{1}{3}$

Probabilistic strategy

Distribution over possible choices

1. Edge *a*: finite distribution
2. Delay for *a*: infinite distribution

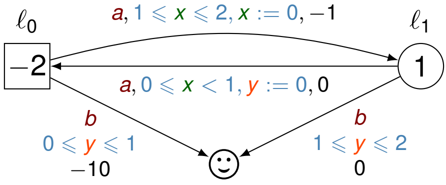
In $(\ell_1, (0, 0))$

Choose between *a* or *b* with $\mathcal{B}(\frac{1}{2})$

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

How to define stochastic strategies?

○ Min □ Max



Deterministic strategy

Choose an edge and a delay

In $(\ell_1, (0,0))$

Choose a with $t = \frac{1}{3}$

Probabilistic strategy

Distribution over possible choices

1. Edge a: finite distribution
2. Delay for a: infinite distribution

In $(\ell_1, (0,0))$

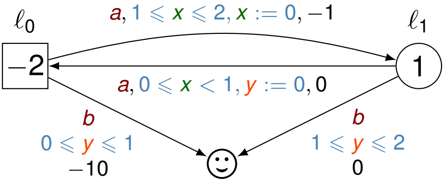
Choose between a or b with $\mathcal{B}(\frac{1}{2})$

- a: choose t with $\mathcal{U}([0,1[)$

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

How to define stochastic strategies?

○ Min □ Max



Probabilistic strategy

Distribution over possible choices

1. Edge a: finite distribution
2. Delay for a: infinite distribution

Deterministic strategy

Choose an edge and a delay

In $(\ell_1, (0, 0))$

Choose a with $t = \frac{1}{3}$

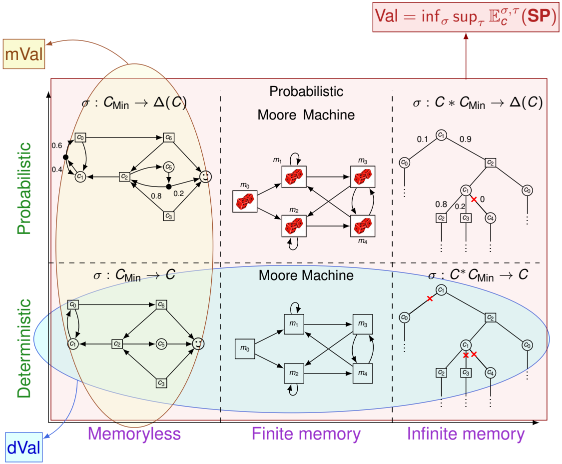
In $(\ell_1, (0, 0))$

Choose between a or b with $\mathcal{B}(\frac{1}{2})$

- a: choose t with $\mathcal{U}([0, 1[)$
- b: choose t with $\delta_{1.5}$

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Summary



For which classes of games?

- Finite shortest path games
- Divergent weighted timed games

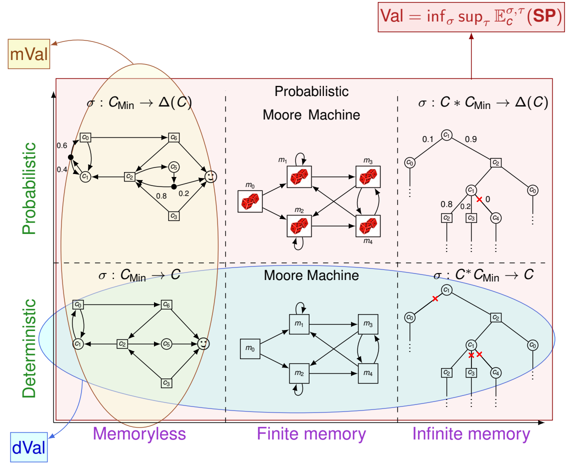
Theorem

Val = dVal = mVal

Summary

For divergent weighted timed games

- Definition of $\mathbb{P}_c^{\sigma,\tau}$



For which classes of games?

- Finite shortest path games
- Divergent weighted timed games

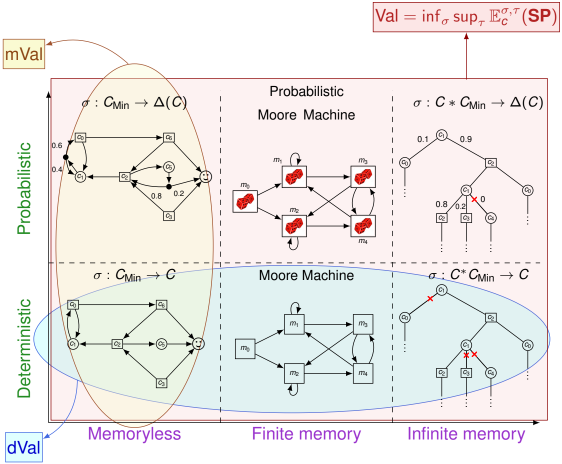
Theorem

$\mathsf{Val} = \mathsf{dVal} = \mathsf{mVal}$

Summary

For divergent weighted timed games

- Definition of $\mathbb{P}_c^{\sigma,\tau}$
- Definition of $\mathbb{E}_c^{\sigma,\tau}(\mathbf{SP})$



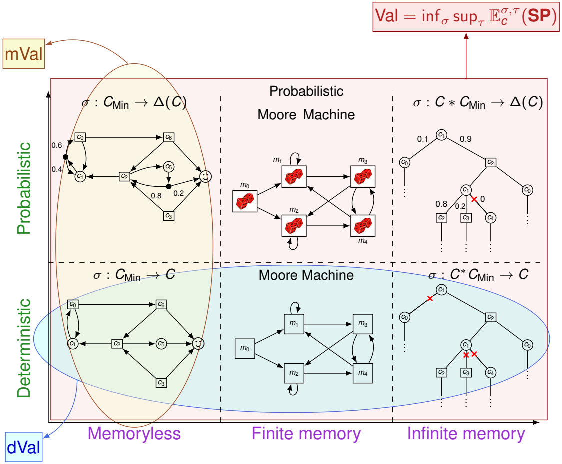
For which classes of games?

- Finite shortest path games
- Divergent weighted timed games

Theorem

$\mathsf{Val} = \mathsf{dVal} = \mathsf{mVal}$

Summary



For divergent weighted timed games

- Definition of $\mathbb{P}_c^{\sigma,\tau}$
- Definition of $\mathbb{E}_c^{\sigma,\tau}(\mathbf{SP})$
- Safety conditions on strategies

For which classes of games?

- Finite shortest path games
- Divergent weighted timed games

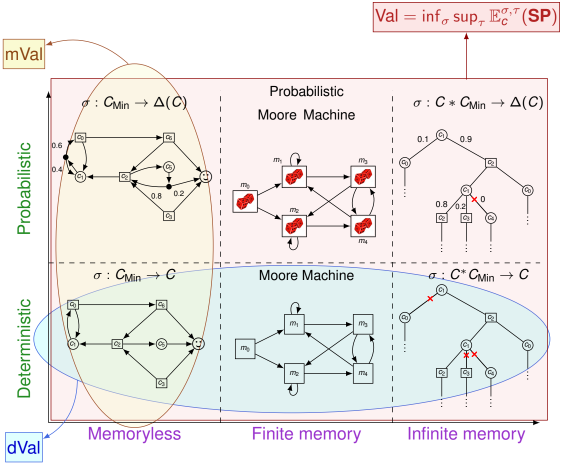
Theorem

Val = dVal = mVal

Summary: perspectives

For divergent weighted timed games

- Definition of $\mathbb{P}_c^{\sigma,\tau}$
- Definition of $\mathbb{E}_c^{\sigma,\tau}(\mathbf{SP})$
- Safety conditions on strategies



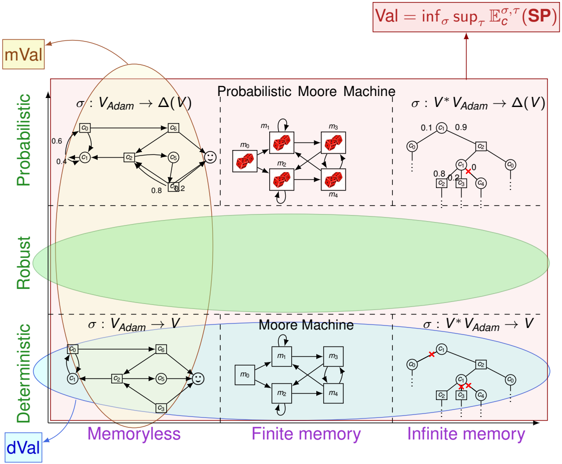
For which classes of games?

- Finite shortest path games
- Divergent weighted timed games

Theorem

$$\text{Val} = \text{dVal} = \text{mVal}$$

Summary: perspectives



For divergent weighted timed games

- Definition of $\mathbb{P}_c^{\sigma,\tau}$
- Definition of $\mathbb{E}_c^{\sigma,\tau}(\mathbf{SP})$
- Safety conditions on strategies

For which classes of games?

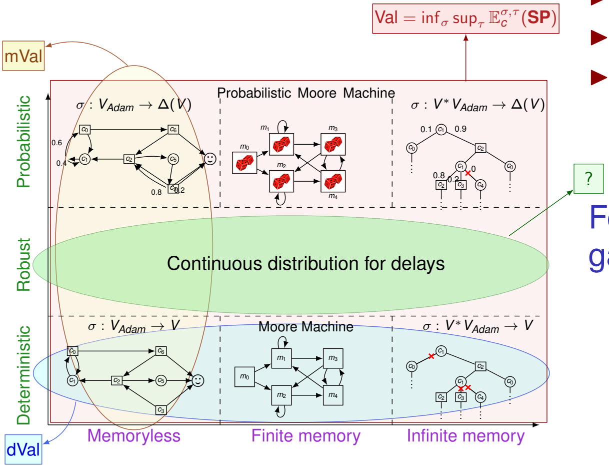
- Finite shortest path games
- Divergent weighted timed games

Theorem

$$\mathsf{Val} = \mathsf{dVal} = \mathsf{mVal}$$

Trading Infinite Memory for Uniform Randomness in Timed Games, K. Chatterjee, T. Henzinger and S. Vinayak, 2008, HSCC

Summary: perspectives



For divergent weighted timed games

- Definition of $\mathbb{P}_c^{\sigma,\tau}$
- Definition of $\mathbb{E}_c^{\sigma,\tau}(\mathbf{SP})$
- Safety conditions on strategies

For which classes of games?

- Finite shortest path games
- Divergent weighted timed games

Theorem

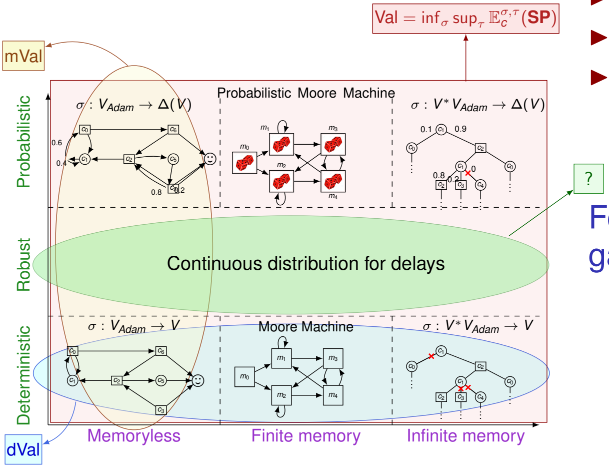
$$\text{Val} = \text{dVal} = \text{mVal}$$

Trading Infinite Memory for Uniform Randomness in Timed Games, K. Chatterjee, T. Henzinger and S. Vinayak, 2008, HSCC

Summary: perspectives

For divergent weighted timed games

- Definition of $\mathbb{P}_c^{\sigma,\tau}$
- Definition of $\mathbb{E}_c^{\sigma,\tau}(\mathbf{SP})$
- Safety conditions on strategies



For which classes of games?

- Finite shortest path games
- Divergent weighted timed games

Theorem

$$\mathsf{Val} = \mathsf{dVal} = \mathsf{mVal}$$

Thank you! Questions?